Towards Capturing Contextual Semantic Information About Statements in Web Tables

Felipe Quécole[1], Romão Martines[1],
José M. Giménez-García[2], and Harsh Thakkar[3]

[1] Federal University of Sao Carlos - UFSCar, São Carlos, Brazil
{felipe.quecole,roh.martines}@gmail.com
[2] Univ Lyon, UJM-Saint-Étienne, CNRS, Laboratoire Hubert Curien
UMR 5516, F-42023 Saint-Étienne, France
jose.gimenez.garcia@univ-st-etienne.fr
[3] University of Bonn, Germany
thakkar@cs.uni-bonn.de

Abstract. Data published on the Web is growing every year. However, most of this data does not have semantic representation. Web tables are an example of structured data on the Web that has no clear semantics. While there is an emerging research effort in lifting tabular data into semantic web formats, most of the work is focused around entity recognition in tables with simple structure. In this work we explore how capture the semantics of complex tables and transform them to knowledge graph. These complex tables include contextual information about statements, such as time or provenance. Hence, we need to use contextualized knowledge graphs to represent the information of the tables. We explore how this contextual information is represented in tables, and relate it to previous classifications of web tables, and how to encode it in RDF using different approaches. Finally, we present a prototype tool that converts web tables from Wikipedia into RDF, trying to cover all existing approaches.

Keywords: Tables, Knowledge Graphs, RDF, Property Graphs

1 Introduction

Data is being published in the web at an ever-increasing speed. However, most of this data lacks semantics. This makes difficult to use it to generate value. Knowledge-graphs are a well-known representation to encode data semantics. The Semantic Web provides standards to represent inter-operable knowledge graphs were each resource can be unequivocally referenced. Tools to generate semantic data from structured web data (specially tables) in gaining traction in the recent years. Most approaches focus on entity recognition and disambiguation, in order to automatically extract the information and transform it to RDF. However, to the best of our knowledge, existing approaches tackle only simple tables with no additional information about the statements that can be extracted. More complex tables exists that provide statements in different contexts (*e.g.*, according to different sources, or valid at different time periods). In order to encode this contextual information (or statement metadata), we need to identify those

Copyright© 2018
for this paper by its authors. Copying permitted for private and academic purposes.

contexts and represent the information accordingly using contextualized knowledge graphs. In this work we focus on transforming tables into RDF, where contexts are represented by means of reifying the statements using the main existing approaches.

The rest of the paper is organized as it follows: in section 2 is discussed some background information; section 3 presents an overview of how data is usually represented in web tables, challenges to represent this data in RDF, and how recent research is dealing with them; section 4 discusses the proposed approach to transform data from web tables to RDF; finally, section 5 draw some conclusions and possible lines of future work.

2 Background

In this section we introduce the necessary background information about RDF, existing reification approaches, and tools to convert automatically structured data to RDF.

2.1 RDF

RDF is the data model used in the Semantic Web. It represents statements as triples `<Subject, Predicate, Object>`. The subject identifies the resource being described, the predicate is the property applied to it, and the object is the concrete value for this property. Triples can share subject and/or object, hence creating a interconnected graph of (possibly heterogeneous) statements. Formal definitions of RDF triple and RDF graph can be seen in Definitions 1 and 2.

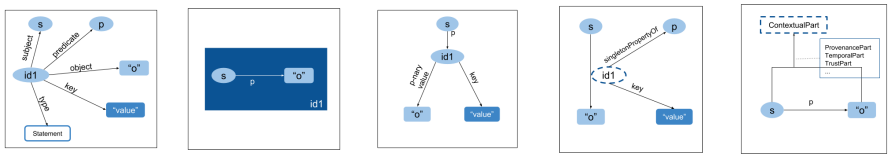
Definition 1 (RDF triple). *Assume infinite, mutually disjoint sets I (IRI references), B (Blank nodes), and L (Literals) . An RDF triple is a tuple $(s,p,o) \in (I \cup B) \times I \times (I \cup B \cup L)$, where "s" is the subject, "p" is the predicate and "o" is the object.*

Definition 2 (RDF graph). *An RDF graph G is a set of RDF triples $\{(s,p,o)\}$. It can be represented as a directed labeled graph $s \xrightarrow{p} o$.*

2.2 Annotating RDF with contextual information

As seen in previous section, RDF statements represent binary relations between to resources (the subject and the object). This model is not well suited to represent additional contextual information about the statement themselves (such as data of validity, provenance, or confidence). Current approaches to represent this kind of information reify the statement into a new resource, that can be then used as subject or object of new statements that represent the context. Down below we describe the five main existing approaches. In the Figure 1, we illustrate each of them.

In RDF Reification [6, Sec. 4], a resource can be used as a statement, and additional information can be added as follows: a quad of the form (s,p,o,i), i is a quad identifier, can be described by the triples $(i,$`r:subject`$,s)$, $(i,$`r:predicate`$,p)$ and $(i,$`r:object`$,o)$.



(a) Standart Reification (b) Named Graphs (c) n-ary Relations (d) Singleton Properties (e) NdFluents

Fig. 1: RDF Approaches

Named Graphs [2] considers a sets of pairs in the form (G,n) where G is a RDF graph and n is an URI (Uniform Resource Identifier). Then, we have N-Quads directly describing an (s,p,o,i) quad.

In N-ary Relations [11], a resource is used to describe a relationship, considering that a subject is involved in a relationship, which in turn has your own identifiers and qualifiers. Here, a quad of the form (s,p,o,i) can be decomposed in (s,p_s,i) and (i,p_v,o), $(p_v,$`:value`$,p)$, $(p_s,$`:statement`$,s)$.

Singleton Properties [10] creates a property that is only used for a unique statement. To represent a quad (s,p,o,i) we need of the triples (s,i,o) and $(i,$`:singlePropertyOf`$,p)$.

NdFluents [5]: creates contextual versions of subject and object and links them to the original and the context using the triples $(s',contextualPartOf,s)$, $(o',contextualPartOf,o)$, $(s',contextualExtent,c)$, $(o',contextualExtent,c)$.

2.3 RDF generation tools

In order to transform a data source into RDF, a common approach is to use a mapping language to represent how the data from one source has to be transformed into triples . Several tools exist to transform heterogeneous data formats into RDF, most of them tackling a single data model or format. In this section we focus on the two most prominent mapping languages: RML [4] and SPARQL-Generate [7]. Our approach will make use of both in different steps of the process.

RML [4] stands for "RDF Mapping Language", it is an extension of R2RML (Relational to RDF Mapping Language) [4]. While R2RML can be used to express customized mappings from relational databases to RDF datasets, RML also supports other structured formats, such as CSV, TSV, XML and JSON. R2RML's mapping references relational tables' column by name, and uses predicates such as SubjectMap, PredicateObjectMap, PredicateMap and ObjectMap. Each of the above mentioned predicates have as object a column or an URI and, the triples are created according to the predicates and their respective referenced column(s). RML extends R2RML vocabulary to include more general clauses (in which the R2RML's clauses are included - as a subset or sub-property), i.e., *rr:logicalTable* and *rr:tableName* become a sub-property of *rml:logicalSource* and *rml:sourceName* In our work, we further

[4] `https://www.w3.org/TR/r2rml`

extend RML to gather enough information from the mapping document and extract from HTML tables, information such as in which column one can find the subject, or which type of table, and thus reification method, is correct for that table, which CSS class should be used to select the specific table from the page, etc.

SPARQL-Generate [7] extends SPARQL 1.1 to be able to extract information from heterogeneous data sources. SPARQL-Generate includes three new clauses:

- **source** clause: used to bind variables to documents
- **iterator** clause: used to extract bits of information from the documents
- **generate** clause: extends the existing *construct* clause of SPARQL 1.1, allowing modularization of queries and factorization of the RDF generation.

The first two clauses (source - and its binding functions - and iterator) allow SPARQL-Generate to support various data formats and navigate through them.

3 Tables on the Web

According to Crestan et al. [3] web tables can be categorized as *layout tables* (used, for presentation purposes, not really containing any knowledge), and *relational tables*. Relational tables encode implicit semantics of the data, and can be further divided according to their structure in *vertical listing*: tables that list in each row one or more attributes for a series of similar entities located in on column (the subject column); *horizontal listing*: similar to vertical listing, horizontal listings present their subjects in one row; *attribute/value*: these tables are a specific case of vertical listings and horizontal listings, but they do not contain the subjects in the table; *matrix*: tables that have the same value type for each cell at the junction of a row and a column; *calendar*: a specific case of the matrix type, differing only in its semantics; and *enumeration*: tables that list a series of objects that have the same ontological relation.

Muñoz et al. [9] identify three types of tables in Wikipedia: *toc*, *infobox*, and *wikitable*. The first corresponds to layout tables, in these tables (and here "toc" stands for: table of content) the topics of the article are presented. The second and the third correspond to relational tables. Infoboxes have a clear horizontal listing structure where the subject, predicate and object of the table can be identified in each row, and form the basis of extracted data to create DBPedia [1]. Wikitables are used to embed tables with semantic content in a Wikipedia article, but their structure is highly variable.

While solutions for transforming data in tables to RDF have been proposed, most of them focus on challenges such as identifying the subject column, interpret the implicit structure of table, entity recognition and disambiguation, and mapping values in the table with classes and properties in a knowledge base [8]. In addition, they only tackle vertical and horizontal listings with simple structure. In this work, we tackle more complex tables, where contextual information needs to be expressed about the extracted triples (such as date or provenance). This contextual information is usually encoded in the tables in one of the following two ways: (1) In horizontal and vertical

listings, by grouping columns by the context[5]. (2) In matrix tables, by using row and column headers as identifiers of the context[6].

4 Approach

The transformation from tables to Knowledge Graphs needs to consider the different typologies of tables presented in the previous section. For tables without contextual metadata about the statements the process is relatively simple: each cell in the subject column is mapped to a subject in a triple and each cell of the same row to an object, using a property that depends on the column of the object. However, for tables that contain contextual information it is necessary to capture the context of the triples. RDF, as mentioned in Section 2.1, only supports binary relations. In order to capture the context of the triples it will be necessary to resort to a reification approach (see Section 2.2). Take as an example table 1[7]. We want to extract information not only about the population estimates, but also about the corresponding year and the agency responsible for that estimation. This table is an example of a matrix table, where contexts are indicated by the headers of rows and columns. Listing 1.1 exemplifies an expected output for the value for the cell of row 1 and column 2, including all the contextual metadata.

Table 1: Subset of World population estimates table from Wikipedia

Year	**United States Census Bureau(2017)**	**Maddison(2008)**
1950	2,557,628,654	2,544,000,000
1951	2,594,939,877	2,571,663,000
1952	2,636,772,306	2,617,949,000
1953	2,682,053,389	2,665,959,000

In addition, the approach needs to read the webpage and extract the information. However, the HTML structure of the table can be arbitrary, and this is one of the challenges to face in this approach. Hence, it is necessary to include a preliminary step to pre-process the table. For this prototype, we decide to get some of the necessary information from the user. The preprocessing step produces as output a modified version of the table with additional information: indexes for column and row, the datatype for the value in each cell, category of the table and groups of columns. This information is then used by a conversion module RDF. Note that this approach could be extended to include other kinds of knowledge graphs, such as property graphs, by adding a new conversion module. A schema of this process is shown in Figure 2.

[5] See `https://en.wikipedia.org/wiki/List_of_sovereign_states_and_dependent_territories_by_mortality_rate`, where the same data is given twice but with different sources

[6] See Table 1

[7] Taken from `https://en.wikipedia.org/wiki/World_population_estimates`

```
wp:year1950  a  time:DateTimeDescription , time:Interval ;
  time:year  "1950"^^xsd:gYear .

wp:Maddison  a  ex:Provenance ;
        prov:wasGeneratedBy [
         a event:Event , prov:Activity ;
         event:time [
          a  time:Interval ;
          time:hasDateTimeDescription  [
            a time:DateTimeDescription ;
            time:year  "2008"^^xsd:gYear ] ] ].

<http://purl.org/az/worldpop#earth:year1950:Maddison>
        rdf:object              2544000000 ;
        rdf:predicate           dbo:populationTotal ;
        rdf:subject             dbr:Earth ;
        time:intervalDuring     wp:year1950 ;
        prov:agent              wp:Maddison .
```

Listing 1.1: Expected output example

The input taken by the preprocessing module is written in RDF using RML [4]. We extend the vocabulary with the following terms:

- *CSSselector*: indicates the CSS selector for the target table in the web page;
- *TablePosition*: index for the target table, given the CSS selector;
- *Reification*: indicates to which category the table belongs;
- *SubjectIndex*: indicates the column that helds the subject for the triple;
- *HeaderRow*: (when columns are grouped by context) indicates in which row the headers (that will be used as predicates) are;
- *ColumnPredicate*: index of the column that is part of the predicate.

The RDF conversion module makes use of SPARQL-Generate [7], using its XPath function to iterate over the elements of the table, and the above mentioned input from the user, except for the first three that are used in the preprocessing step, are used to compose the SPARQL-generate query. The values inserted by the user dictate the role for each column from the HTML table, that is, which column is the subject, part of the predicate or just the object of the triples (with the header being the predicate).

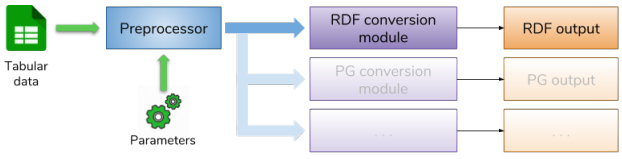


Fig. 2: Table to KG transformation workflow

The prototype tool is publicly available[8] under Apache-2.0 license.

[8] https://github.com/felipequecole/table2rdf

5 Conclusions

Transforming web tables into knowledge graphs while capturing their semantics and contextual information is a challenging task for various reasons: On the side of the knowledge graph representation, it can be necessary to use reification techniques in order to encode the context. On the side of the table, the HTML structure can be arbitrary, and the contents of the table can be difficult to identify. We propose a two-step process. The first step takes additional information and pre-processes the table, generating a enriched version of the table with the information needed by the second step, such as the category of the table or how to extract the contextual metadata about the statements. The second step reads the output of the preprocessor and transforms the data in a knowledge graph. We have implemented a tool that gets part of the necessary information from the user (falling back to default values in case some information is not given) in the first step, and a RDF conversion module as second step. Note that other approaches focusing on different challenges, such as entity disambiguation or subject column identification, could be incorporated in the preprocessing step. Conversely, new modules can be added to substitute the RDF transformation to another kind of knowledge graph, such as property graphs.

Acknowledgements: This work is supported by H2020 Marie Skłodowska-Curie ITN No 642795.

References

[1] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. ISWC+ASWC (2007).
[2] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. J. Web Sem. (2005).
[3] Crestan, E., Pantel, P.: Web-scale table census and classification. Proceedings of the fourth ACM international conference on Web search and data mining (2011).
[4] Dimou, A., Sande, M.V., Colpaert, P., Verborgh, R., Mannens, E., deWalle, R.V.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. LDOW (2014).
[5] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. ESWC (2017).
[6] Lassila, O., Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation (1999).
[7] Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL Extension for Generating RDF from Heterogeneous Formats. ESWC (2017).
[8] Martinez-Rodriguez, J., Hogan, A., Lopez-Arevalo, I.: Information Extraction meets the Semantic Web: A Survey. Semantic Web journal (2018).
[9] Muñoz, E., Hogan, A., Mileo, A.: Using linked data to mine RDF from wikipedia's tables. Proceedings of the 7th ACM international conference on Web search and data mining (2014).
[10] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. WWW (2014).
[11] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web. W3C Working Group (2006).